

Project Implementation
COS 301 Buzz Project
Group: Testing Phase: Resources
Version 1.0

Carla de Beer 95151835
Prenolan Govender 13102380
Shaun Meintjies 13310896
Collins Mphahlele 12211070
Dumisani Msiza 12225887
Joseph Murray 12030733
Sifiso Shabangu 12081622
Joseph Potgieter 12003672
Johan van Rooyen 11205131

<https://github.com/Carla-de-Beer/Testing-Resources>

University of Pretoria

24 April 2015

Contents

1	Introduction	2
2	Functional Testing	2
2.1	Resources A	2
2.1.1	Pre-condition violations	2
2.1.2	Post-condition violations	2
2.1.3	Data structure requirements	2
2.1.4	removeResourceUseCase	2
2.1.5	removeResourceUseCase	3
2.2	Resources B	3
2.2.1	Pre-condition violations	3
2.2.2	Post-condition violations	3
2.2.3	Data structure requirements	3
3	Non-functional testing / assessment	3
3.1	Resources A	3
3.1.1	Usability	3
3.1.2	Performance	3
3.1.3	Scalability	3
3.1.4	Testability	3
3.1.5	Security	3
3.1.6	Reliability	3
3.1.7	Reusability	3
3.1.8	Pluggability	3
3.2	Resources B	3
3.2.1	Usability	4
3.2.2	Performance	4
3.2.3	Scalability	4
3.2.4	Testability	4
3.2.5	Security	4
3.2.6	Reliability	4
3.2.7	Reusability	4
3.2.8	Pluggability	4
4	Critical Evaluation and Recommendations	4
4.1	Resources A	4
4.2	Resources B	4

1 Introduction

The purpose of this task was to test the functionality provided by the Resources teams for the Buzz Project.

According to the master specification document, version 0.1, released 13 March 2015, the buzzResources module is used to upload and manage resources like media (e.g. images, video) and documents (e.g. PDF documents, Open Document Format documents). These resources are to be either embedded or linked to in posts.

This team took each of the pre and post-conditions of the required use case, considering the work of both Resources A and Resources B, and tested them for compliance with the functional requirements.

The use case for this team is called **uploadResource**. In terms of the requirements for this use case, and as defined by the master specification document, users should be able to upload resources such as media files or documents. Any uploaded resource should be accessible by other users who should also be able to specify links to that resource.

The functional requirements preconditions for this use case included the need to

- detect the mime type
- check that size constraints are met
- check that the resource type is supported

The functional requirements postconditions for this use case included the need to

- check that the resource persisted
- check that the URL for the resource was created

2 Functional Testing

2.1 Resources A

2.1.1 Pre-condition violations

2.1.2 Post-condition violations

2.1.3 Data structure requirements

2.1.4 removeResourceUseCase

For this use case, no contract requirements were specified. It is assumed that a pre-condition for this use case would be that the database contains the ID being passed as a parameter to removeResource. The post-condition would

be the removal of a resource entry from the database.
removeResource successfully conforms to assumed pre-conditions and post-conditions. The pre-condition is tested before any manipulation of the database is performed. The post-condition (removal from the database) is properly reflected by the database.

2.1.5 removeResourceUseCase

No problems detected with regards to the bigger picture in terms of the system due to this use case being relatively simple.

2.2 Resources B

2.2.1 Pre-condition violations

2.2.2 Post-condition violations

2.2.3 Data structure requirements

3 Non-functional testing / assessment

3.1 Resources A

The use case was tested against the following list of architectural requirements:

3.1.1 Usability

3.1.2 Performance

3.1.3 Scalability

3.1.4 Testability

3.1.5 Security

3.1.6 Reliability

3.1.7 Reusability

3.1.8 Pluggability

3.2 Resources B

The use case was tested against the following list of architectural requirements:

3.2.1 Usability

3.2.2 Performance

3.2.3 Scalability

3.2.4 Testability

3.2.5 Security

3.2.6 Reliability

3.2.7 Reusability

3.2.8 Pluggability

4 Critical Evaluation and Recommendations

4.1 Resources A

4.2 Resources B